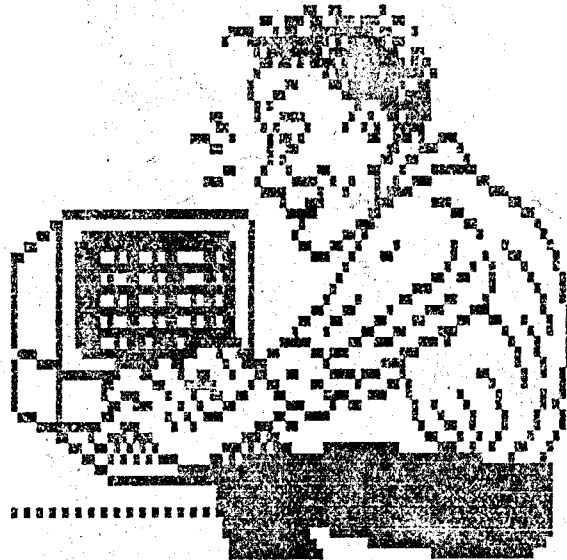


QL TECHNICAL REVIEW



EDITORIAL & NEWS	2
THOR SCENE	3
QRAM REVIEW	4-7
MAKING A QL MONSTER	8-14
BITS AND PIECES	14&23
TOOLKIT II REVIEW	15-16
MACHINE CODE TUTORIAL (PART 1)	17-23
IDIS DISASSEMBLER REVIEW	24-25
REPAIRS FOR THE QL?	26
CP/M 68K ON THE QL	27-30
LIGHTNING AND SPEEDSCREEN REVIEW	31-32

SUBSCRIPTIONS £5.00 FOR 4 ISSUES FROM C.G.H. SERVICES, CWM GWEN HALL, PENCADER, DYFED, CYMRU. SA 39 9HA.

EDITORIAL

NEWS

Not a lot to report here that I have first hand experience of. Obviously Miracle and Rebel Electronics' hard disk systems will appeal to those wanting to store large amounts of data. Probably this will make it possible to at last run bulletin boards on the QL and also more PC software. Maybe people will take the leap from mdvs to hard disks and miss out floppies altogether? If you have a hard disk then let us know how you're getting on with it.

QUANTA - to which I hope you all belong - have recently been doing some excellent bargain deals for members, including Jan Jones definitive work on SuperBasic and the Tandata modem stacks. In the light of which I'd like to propose a Comms column for QLTR. In particular let's nominate a board where we can all congregate. (QLSUB?)

On the MS-DOS Emulation front, I understand that the ANT Computing one still isn't up to scratch, whereas The Solution does at least work with a lot of the software - even if it is at a snail's pace. I'll write a review of The Solution for next issue (assuming I can find a few spare evenings) - and I'd like to hear other people's opinions as well. The Schon hardware emulator sounds fun - but if the device is too expensive it'll be cheaper to get second-hand PC.

Strong Computers of Carmarthen are no longer handling the ST QL emulator. By the sound of a recent article in QUANTA this is fast, running QL software at 2 or 3 times QL speed. Jochen Merz Software of Duisberg in Germany are now selling emulator boards at £180. If you have an ST and need to run QL software as well then it's a useful purchase. (Transferring files between the computers is now very simple using the ST-QL file transfer utility in the Public Domain from Germany and our home-grown (courtesy of Alan Pemberton and Rich Mellor) ST-QL screen transfer prog.

Also on the ST front (damned versatile computers) Tony Tebby has been demonstrating his new QL compatible operating system for the ST. (How many people know that under the GEM interface lies CP/M 68K on the ST? Well that's what one of the books about the ST says!) Finally the Minerva QDOS replacement is on sale. Apparently it speeds things up and fixes many QDOS bugs.

It has taken a lot longer than I expected to get this issue of QL Technical Review out. Let's hope it doesn't take another year to do the next issue. This magazine grew out of QL Adventurers' Forum and the work several of us had been doing on CGH Services' products. We reckoned that the experience we had gained and the comparative value of the products we use may be of interest to other QL users. If nothing else it opens up another channel of communication for QL Users.

Although some of the contributors are programmers, most of our readers we expect to be users and hobbyists and it is mainly to them that this magazine is addressed. Having said which, please don't think we're excluding anyone. This magazine is open to anyone who uses their QL.

We have already been promised several articles for Issue 2 of this mag, probably people were waiting to see if Issue 1 ever materialised! So expect reviews of Text87, QLiberator (latest edition), and a short item or two on networking, Rich Mellor's second episode on machine code programming and so forth. (Forth?) Whether your speciality is hardware modifications, languages and programming, using specialist software or if you're just plain curious about what you can get a QL to do then write in and let other people know.

We'd also like to operate a Helpline service in this mag. I have to confess to knowing even less about the technical side of QL's than adventures and games but that again is where you, our readers come in. If you can help please write in and share your knowledge.

Due to the gestation time of this journal, some of the articles may seem a little dated. For example there is now a new special edition of Lightning which is, I presume, much better than the old one and so even much more better than Speedscreen.

Any way, enough rabbiting for this issue - time for you lot to get reading and shortly, I hope, to get writing too. Let's set the deadline for Issue 2 at November 1st - it'll give us all something to aim at.

Richard

THOR SCENE

Malcolm Smith has sent the following letter to enlighten us on current developments. Please note that all opinions are Malcolm's. I don't have a THOR XVI (and am unlikely to be able to afford one) so am unable to verify any of Malcolm's statements. Which is not to doubt him in any way as he has been more than helpful in testing CGH Services products over the past year. If DP or CST care to comment on this article we shall be pleased to print their response.

"Well CST seem to be doing more than alright. They appear to be selling all they make and because the Danish market currently demands such machines they are concentrating on that for the time being. However, they do expect to start on the U.K. market pretty soon and that's when we ought to see English manuals and a few reviews in the U.K. press.

Only recently have they been in real contact with the Software Houses in the U.K. I know for a fact that Digital Precision now have a THOR XVI and they are trying out their programs on it. However, the latest ROM version of the Argus Operating System (v6.40) seems to have a few bugs in it. This means that a pile of DP software (including EDITOR) won't run. DP have reverted to the 6.39 version and they seem quite happy with that.

I have the 6.40 ROM in the machine as I write and it copes well with Text87 software. As an indication of the helpfulness of CST they went out of their way to recommend this package. My only regret is that I didn't find it earlier. I had a copy of the 6.39 ROM but I swapped it for the 6.40, but somewhere in my desk I have a copy of 6.37. I can understand that there are a few bugs in this version as the "feel" of Argus is different. For example, we no longer have the spotty screen on boot-up as the RAM is checked. Instead there is a very pleasant yellow screen. There are also a few more commands available, for example SAVE_0 is now implemented, which wasn't on 6.39. The problem with 6.40 is that the EDITOR doesn't work at all, (it just crashes), and one of my games, DOMINATION, has its job removed as soon as the game is about to start (after the players' names have been entered). This is a real shame as I like to play this odd little game as a form of relaxation.

CST have also started to produce their own software. I have bought off them their ARCTurus editor and I must admit I find it superior to DP's Special Edition Editor. I find it faster to use (the text handling certainly is faster!) and it multitasks within itself. In other words, one ARC'ed job can have a number of editors running under it, so if one wishes to use more than one editor one needn't call up the editor again and use more memory as in DP's case. One simply calls up another editing job. Another neat feature is that pipes are able to be created between these jobs and so one can move/copy blocks of data between jobs. An excellent idea.

I find CST and Digital Precision both helpful and understanding. Now that communication seems to have improved between these two companies things ought to look brighter on the THOR XVI front. Employees of both companies have been extremely helpful with my queries. I am getting more and more confident in the THOR XVI scene here in Europe.

All I wish now is for a new ROM (or Turbo Toolkit) to appear which will make all my compatibility problems disappear and then I'll be the happiest computer user in Norway. Will the THOR XVI ever be a serious challenge to the IBM PC? I would like to think so, but it would need a faster emulator than SOLUTION to make it so. Oh yes, SOLUTION does work, but it's rather slow and it can't really compete with the AT's that hang around the office for speed. And worse! I can't seem to get ELITE to run under it!"

Malcolm Smith would be very interested to hear from any THOR XVI owners out there (and earlier THOR's as well - they tend to get even more neglected!) so write to him at: Statsråd Ihlensvei 66b, N-2010 Strommen, Norway. (Also any Scandinavian QL owners - pass the message on.)

Also QLTR will be very pleased to hear from CST, DP and other software houses of their progress in bringing about the successful development of the THOR XVI and suitable software to run on it. Hopefully when the ROM is stable other people will start writing with the THOR in mind. English manuals would also be a distinct advance too.

Richard

A REVIEW OF QRAM, QTYP AND THE HOTKEY II SYSTEM
BY QJUMP LTD., 24 KING STREET, RAMPTON CAMBRIDGE CB4 4QD

QRAM (£29.95)

This is not only another front end program, but also helps to provide the QL with another operating system. It gives non-destructive windows for tasks EXECuted whilst QRAM is present. This means that the screen occupied by one task will not be over-written by another, since when CTRL C is pressed (to switch between tasks), the current screen is saved. Unfortunately, this can lead to problems with some programs, since QRAM only replaces the windows which are still open. However, QJUMP have foreseen this, and have included the facility to 'unlock' a task and give it the normal destructive windows of the QL (ie. the screen is not cleared when a task is complete).

In contrast with some other similar programs, please note that if CTRL C is used, QRAM allows the task to continue in the background as long as it does not attempt to write anything to the screen, in which case the task will be suspended until control is returned to it.

The package includes a new display driver which makes the MODE command more reliable: no longer are the CONsole screens altered to black when a MODE command is issued. Also the display driver means that SPEEDSCREEN (Creative Codeworks) no longer causes the screen to flash when the mode is changed, but not the PAPER and INK colours.

As for the actual front end system itself, QRAM enables fast and selective directories, copying, viewing and deleting of files from any device, with full wild card facilities. Tasks can be executed, suspended, removed or re-activated but unfortunately, the user is unable to alter the priority of a task which seems quite an omission.

Also included is an excellent screen dump which allows the dot density to be altered and a screen (or window) 'snapshot' to be taken. Another facility is the ability to list (and close) all of the channels currently open along with the program which opened them, thus making it easy to track down an elusive 'IN USE' error.

The 'options' menu, lets you configure QRAM as you like, either permanently or just until the QL is reset. You can even alter the mouse sensitivity according to how far you wish to move the mouse to have the desired effect upon the screen. This is a very useful ability which most other (if not all) packages omit. Do not worry if you do not possess a mouse, since QRAM works just as well with the keyboard.

Lastly a very useful ram disk and print spooler are also included in the package.

Overall, the package is a most worthwhile investment if you have expanded memory (note QRAM needs at least 256K RAM), and after a few hours use of it, you will no doubt wonder why anyone could ever have second thoughts about buying it.

HOTKEY II SYSTEM

This is currently only supplied with QTYP version 2 and QPAC -1 (also available from QJUMP).

The HOTKEY system was first provided with QRAM, but the earlier version required the user to select which programs to 'hotkey' and then merged them all into one large file which had to be loaded when the QL was reset and then could not be altered. New additions allow the user to select or deselect which hotkey calls which task by the

use of simple keywords. Another useful difference is that HOTKEY II recognises the hotkey in both upper and lower case, since this could cause problems in the earlier version.

This utility lets the user not only provide a hotkey for a task (a) but also to assign a key to a task so that once it is installed, the user can switch between tasks by merely pressing ALT and the designated key. This eliminates searching through every task present on the QL by repeated presses of CTRL C, and also means that a separate switching program is not needed.

Indeed, the user can even specify a hotkey to load and execute a task, if it is not used often enough to justify making it resident in memory.

Other functions added to Superbasic are the ability to recall the last line entered and to 'stuff' a string of characters into the input buffer. (b). However a utility which would have been useful would have been the ability to start a task with specified characters in its command window, so that for example, you could auto-start QUILL with the margins already set and a certain document loaded.

The user can also use the system to set a GUARDIAN window, which will save a defined area of the screen, rather than just that used by the job itself. The PSION programs can also be 'tamed' by the use of the GRABBER function which will allot a certain amount of memory only for the programs, thus allowing them to multi-task alongside any other program. Note that both of these facilities were also provided with the current version of QRAM, but this meant altering the actual code of the program on disk. Now only the copy of the program in memory is altered, thus enabling the PSION configurator to be used since there is no extra code tagged onto the front of the program.

All of these utilities can be used without actually having to 'hotkey' the programs, by using the new command 'EXEP' instead of the normal EXEC.

Thankfully for those of us without 'JS' ROMS nearly all of the extra facilities are added to superbasic as functions which will return an error if something goes wrong. To make use of this facility, QJUMP have added the command 'ERT' which will print the error message returned to #0.

Overall this part of the QRAM environment is very well written, and although it possesses a few minor quirks (c), it is a useful addition to the range of facilities for the QL and will enable even the newest user of the QL to utilise the multi-tasking facilities to the full. To this end, a few example BOOT files are included to hotkey the PSION programs and EDITOR (Digital Precision).

QTYP (£29.95)

This is a real-time spelling checker for the QL with a dictionary of over 45,000 words (although due to text compression, the actual dictionary only takes up around 75K. - Why not let us have this compression program separate QJUMP ?). As you can see from the above statement, dictionary size means that QTYP is only of any real use to users with expanded QLs.

The checker is invoked by pressing ALT t (user configurable) and checks each letter you type against its dictionary to see if you are typing a valid word. If there is any deviation from the dictionary words, a warning beep is sounded instead of the normal "all's well" beep. If you continue to type the word, when a non-letter (d) is typed, QTYP will ask if you want to add the word to a temporary word list (e) to merge into the main dictionary later. It is not necessary to merge the temporary word list at the

end of every session since QTYP will automatically load it when it is invoked, along with the main dictionary.

Thankfully, QTYP does not visibly slow the QL down, and switches off automatically if you use the cursor keys beyond the word being currently typed. Do not worry though about this, since you do not have to turn QTYP on again - the system is switched on again as soon as a non-letter is typed.

Another feature of QTYP is the ability to display a message window for problem words which can have one of two spellings according to the circumstances. The message can be specified by the user by entering it after the problem word into the dictionary.

Unless you are actually in the process of writing something, the above will be of little use to you, so to provide for this, QJUMP have provided the utility 'QTYP_FILE' which will check the spelling in a currently existing document or file. This is a very useful facility considering the amount of spelling errors that can be found in any number of adventures on the QL - I just wish that some of the software houses would use this product and other similar ones before they let the programs loose on us, the unsuspecting public.

When using Qtyp_file, the user can decide whether to add all words which are not in the dictionary to a temporary word file, or to just set a little marker against them. Unfortunately the latter option can lead to a difficulty in that the marker will overwrite the character immediately after the word unless it is a space.

This facility is very useful for another aspect of the QTYP system - that of setting up specialist dictionaries containing words you normally use for one certain type of application, such as BASIC programming (the variables aren't always sensible words, even if written by the best of programmers).

The setting up of specialist dictionaries is very useful, especially if coupled with another facility of QTYP, which is the ability to use new keywords to control the checking of words typed in your own programs. It is a shame that no-one has bothered to incorporate this feature into a text editor or similar program commercially. The manual includes some clear examples of how to use this facility in both BASIC and machine code, so all of the necessary information is there, just waiting for the right programmer to come along and use it.

QTYP is yet another very useful program from QJUMP and is not for just those users who use the QL as a word processor. It is especially helpful for those of us who aren't very good at spelling since the program will give a list of words which have the same beginning as the letters you have already typed and if necessary enter the word at the desired place. The market for this program has been made even wider by the fact that QRAM no longer has to be present to run the program.

CONCLUSION

All three of these packages make the QL into a very powerful machine and are becoming more and more useful as the amount of memory available to the QL increases. With talk of a three megabyte expansion coming out soon, it would indeed be a waste if the user was only to use one program at a time and not have three or four (maybe even more) programs running concurrently.

Another interesting point is that all of the Superbasic extensions use a special piece of code (which Tony Tebby is willing to part with if your nice to him) which enable them to be used in the same program as that which invokes them. For example the following could be used:

10 a=RESPR(100):LBYTES mdvl_exep,a:CALL a:REMark add command EXEP to basic
20 EXEP mdvl_task,u:REMark EXECute task with QRAM's display driver turned off

The manuals are all excellent and very clearly written (no searching through 100s of pages to find one bit of information) and all of the disks contain backup copies of all the programs just in case you fail to follow QJUMP's advice and make a backup copy yourself.

It is a pity that QJUMP have now decided to withdraw their advertising from QL WORLD (for various reasons) since this seems to be the main source of information for QL users. However, not to worry, this does not mean that QJUMP are contemplating leaving the QL market, since they now produce their own newsletter and there is talk of a much improved QRAM II which will include many more facilities and also utilise the new HOTKEY II system to full effect so that less of the QL's memory will be used.

Programs are now beginning to appear on the market which use QRAM to its full potential, including a menu driven text editor, although unfortunately for many QL users, much of the new software is written in German and may take some time to come across to Britain. Now that QJUMP have released a package (QPTR) which enables authors of software to use QRAM's facilities in their own programs and also basic programs using it can now be compiled using LIBERATOR (Liberation Software) (f), let us hope that a lot more software will be forthcoming soon.

NOTES

a) Under the original HOTKEY system, TURBOcharged and BCPL programs couldn't be assigned to a HOTKEY since they do not use 'pure' code (code which can be put into a ROM). HOTKEY II now overcomes this problem by the user adding the parameter 'i' (for impure code) to the hotkey commands. However, please note that QJUMP warns that this will eat up the QL's memory, since two copies of the code are made to ensure that the original code remains unmodified.

b) Both of these facilities are also present in the SUPERTOOLKIT II but these get rid of a few hiccups which may occur under the latter implementation.

c) One of the quirks, which I have now notified Qjump, of is the fact that if a compiled program sets up a hotkey, and the task is then removed before it is able to remove the hotkey, any attempt to re-assign that hotkey from BASIC (or another task) will fail until another hotkey has been removed.

d) QTYP recognises all letters (English and foreign) that can be entered on the keyboard but ignores any punctuation at all except for abbreviation marks. Note that this is different from the earlier versions which would not recognise abbreviation marks and would thus ask you if you wanted to add 'wouldn' to the dictionary if you tried to type "wouldn't".

e) The word which is not present in the dictionary is not added to the main dictionary straight away so that you can enter any plurals at the same time.

f) Turbo (Digital Precision) is at the present time unable to compile programs using the QPTR interface due to the fact that it is not a complete Superbasic compiler (Qjump's words, not mine). However this is now being worked upon and should be changed within the next few months. However it must be noted that this would involve a major re-write and Qjump fear that it will never be completed due to the low volume sales of Turbo.

Rich Mellor

CREATING A QL MONSTER - BY MICHAEL L JACKSON

I have owned a QL for three and a half years, having taken advantage of Dixon's QL and 8056 printer deal back in 1985. For most of the time I have also been a member of Quanta (I still am), and have been fascinated by the numerous articles about various modifications QL enthusiasts have made to their basic 128K keyboard-size computers. I soon added an extra 512K RAM to my QL, a Sandy Thru-con expansion card. When scare stories about an end to microdrive production began circulating after Sir Clive's Amstrad deal, I took the plunge and purchased a 3.5 inch disk drive and interface.

THE SINCLAIR DISK DRIVE CAMPAIGN

This latter action is worth several chapters in its own right, at least as far as I'm concerned. I bought a Sinclair-badged Micro Peripherals drive outfit from a well known chemist chain for £150.00, compared to an original selling price of about £240.00. I thought I had got a bargain, though admittedly its specification was poor compared to more modern units with QJump interface ROMs - the Sinclair interface came with a number of utility programs on 3.5" disk to make up for the lack of decent toolkit commands on ROM. After about three months moderate usage the drive stopped working correctly. It was returned to the supplier, and then began a six month battle to get it repaired under guarantee or to be given a refund. Shortly after I had a letter published in Popular Computing Weekly I received a full refund, plus a couple of gift vouchers to make up for the numerous journeys to the shop, a round trip of 10 miles a time.

With the refund I purchased a Sandy Superdisk interface and 3.5 inch drive. At £180.00 it was a bit more, but a better specification than the Sinclair model and included a parallel printer port (device PAR) and better toolkit commands. I had a few initial problems, which Sandy sorted out promptly. My main contact was Arnie Gardner, who even sent me an extra drive and interface on trust until I got a combination which worked correctly with my QL. His willingness to help persuaded me to have Sandy upgrade the machine to a JS ROM set. They returned the QL after giving it a thorough test and tweaking a bit better performance from the microdrives. Sandy PCP's departure from the QL scene is a sad loss because their products seemed to be among the best. Certainly I have had no problems with my Sandy hardware.

AN ADDITIONAL QL

Once I had got a decent QL set up working reliably I began to feel the urge to be a bit more adventurous. (To be honest, I used to think that fitting a disk interface was adventurous, but then I had no experience of playing around with computer hardware.) I started examining the pages of Micro Mart, a fortnightly equivalent of Exchange and Mart but specialising in computer hardware and software from Acorn Archimedes to ZX80. I thought a back-up QL would be useful, though not vital because the temporary loss of the QL would affect my well being but not my wealth. From a Southport teenager moving over to an Atari ST I bought a second hand QL and Tandata QConnect modem, plus a few books and magazines.

I was fairly pleased to discover that the QL was a JS issue seven board, which I believe must have been one of the last ones manufactured in the United Kingdom for the British market. It had not taken hard punishment from its previous owner, but the keyboard had a few keys with rough edges which tended to distract while typing. Fortunately it was at that time that Schon halved the price of their standard replacement keyboards to £35. Having read favourable reviews in Sinclair QL World and Quanta I summoned up courage and sent off for one.

REPLACING THE KEYBOARD

The Schon replacement keyboard is quite an impressive product. It differs slightly from photographs published, for instance, in Popular Computing Weekly in that the lettering on some keys has been changed. The function 'F'-keys' are in red, the alphabetic and cursor keys in dark grey, and the numeric and other keys (SHIFT, CTRL, ALT, etc) are black. The lettering is white in each case. Regretably the +/- key is fitted upside down on my keyboard, and several other owners have reported the same defect. (I wrote to Schon about this, enclosing a SAE, but never received a reply.) The keyboard is a complete replacement for the QL top, so bid farewell to the bubble mat and membrane. It takes longer than the advertised five minutes to fit, allowing for spells of panic and brow-mopping, but is not too difficult and the instructions supplied with it are straight forward. Previously published articles in QL World and Quanta give further hints for the faint-hearted. I would say the main problems are replacing the LED leads for the power and microdrive lights (but Schon colour code theirs better than Sinclair) and fitting the dreaded anti-bounce chip. I say dreaded because this chip, which is a customised 8749 to replace the 8049 keyboard controller, has an unfortunate bug for modem users. The new chip improves keyboard performance so that your QL will rival machines costing many hundreds of pounds more, but it interferes with the use of the serial ports (SER2 at least). My original QL remains unmodified so I was able to use the Tandata modem to log on to the increasing number of QL bulletin boards. Other purchasers of the anti-bounce chip had discovered the problem and details of the bug were revealed on several BBs, so I realised it was not caused by the modem.

Incidentally, it has been written elsewhere that the Schon chip cures the CTRL<ALT<7 bug, a keypress combination which locks up the QL, thus requiring a reset. This is not the case - the bug is still present with the Schon chip. The only cure for that bug that I have tried successfully is a little machine code pop-up utility by Steve Jones on the Forum #3 microdrive magazine.

THE HEAT PROBLEM

At the same time as buying the new keyboard, I followed some Quanta advice to release the heat which builds up inside the QL behind the microdrives. Overheating caused me some problems in the form of lock ups in the first summer I had the QL. At first I didn't understand the problem, but I soon got the message as overheating seems to be a favourite topic among QL owners. On my first QL the best solution seemed to buy one of Dennis Briggs' replacement power supplies. These have a nice big heat sink on the outside, and prevent even a fully expanded QL from getting warm. While I had got the top off the other QL, to fit the Schon keyboard, I tried the suggestion of drilling a grid of holes in the casing behind MDV2 - i.e. above the internal heatsink. I did the same with the original Sinclair power supplies. This hole drilling (done very carefully) does help to an extent, but is not the complete solution. In the recent hot weather I have been using both QLs at once (operator multi-tasking?) and one locked up from time to time, the other kept going despite everything around it appearing to melt.... The one that didn't crash has the Briggs power supply. These are definitely worth trying if you have a heat problem, and cost about £16 compared to the £25 usually asked for the inferior Sinclair power units.

IMPROVE YOUR NETWORK WITH TK2!

Now having two QLs I had the opportunity to try out the network at my leisure. At first I found the commands as described in the QL User Guide a bit vague. I had read that the QJump Supertoolkit II by Tony Tebby improves networking commands, and as I had it on the disk interface of one QL, I decided to order it for the other to plug in the ROM port. Please note that TK2 is needed on ROM for its networking improvements to work - the disk or cartridge versions are not sufficient (nor are copies of the ROM loaded into RAM - I tried it). Needless to say, TK2 is worth having for the numerous extra commands and bug-fixes, but its

network improvements made it a must for me. I was still using the original short Sinclair network lead until recently, which meant that the QLs had to be side by side. However, when my pet hamster escaped overnight and chewed through my hi-fi speaker leads I thought I would ask the local High Street electrical shop people if they could make a 5 metre network lead as well as repairing the speakers. This caused them no problems - the lead was microphone wire with a mono 3.5mm jackplug on each end, which worked first time. The longer lead enabled me to rearrange the QLs into more satisfactory positions, and if it has caused any deterioration in network performance I haven't noticed it.

Computer Shopper carried some bad news about the MGT SAM 'super Spectrum', which is said to have a built in network. Though the SAM is supposed to be more Spectrum-compatible than the Amstrad Spectrums, and also offers many improvements over the original design, its network will not work with QLs and Spectrums. If it had done so this would have been a big selling point for me.

AN AQUAINTENCE WITH QUEST

During the past year I have bought a number of products once sold by Quest International Computer Technology Limited of Hampshire, an early supplier in the QL hardware market c1984-85. Quest had dropped out of the market by the time I bought my first QL.

I was interested in extra memory for the second QL, but not a full price new product because my main QL already had 640K RAM. I saw nameless memory expansion boards advertised in Micro Mart and QL World, and the prices seemed to be falling at a time when world RAM chip prices were rocketing skywards. I could hardly believe it when the price fell to £25.00 for 256K, which would increase the memory of the second QL to a respectable 384K. The supplier confirmed that the boards were brand new, but surplus stock, and lacked a through port. (The through port was a big selling point of the Sandy Thru-con and the Miracle Systems Expanderram, both now unobtainable apart from second-hand.) I sent off my cheque, not quite knowing what to expect.

The parcel I received was rather bigger than I expected, having had experience only of the tiny Sandy RAM boards. The 256K was a Quest expansion, which according to an old Sinclair User guide cost in 1984 over ten times the price I had paid in 1988 (presumably because it had little competition in the early days of the QL). The new look to the packaging and the enclosed (useless in 1988) guarantee card seemed to confirm that the product was unused. The expansion was housed in a large metal cover which added a good few inches to the length of the QL. The 256K was probably more than three times as long as the 512K Sandy expansion, a much more modern design. The Quest board used older, slower chips. Using a RAM speed test published in Computer Shopper's QL column I confirmed that the Sandy external expansion was about 60% faster than internal RAM, but that the Quest RAM was probably only around 13% faster.

QUEST EXPANSION CONSOLE

Despite my realisation that Quest's memory products were a bit old fashioned by modern QL standards, I was still pleased to see an advertisement at the back of Quanta for expansion consoles made by the same firm. I had become concerned about the length of my main QL, which had a disk interface plugged into a memory expansion. There seemed a lot of potential for wobble, though the QL never crashed for that reason, and there was a lack of prospect for future expansion. Some sort of expansion box seemed desirable, but there was little on offer on the market. Sandy had stopped production of its QXT640 kits, which were rather too expensive for me and were said to have some problems with software which used the KEYROW command because of the new keyboards provided with them. The SPEM QL Personal System/2 kits sounded interesting but again are expensive and supplies from Italy are reportedly sporadic. CST had fled to Denmark, but had already stopped production of their Q+4 units. The Schon boxes sound beyond me. I assumed

therefore that I had little hope of further expansion, until I saw the advertisement in Quanta.

The idea of a four slot expansion box for about £40 seemed excellent, so I ordered one from a Quanta member who had a small number of them salvaged from Quest. It arrived in an enormous box, packed in a lifetime's supply of foam squiggles (which proved useful for insulating my dog's kennel). No instructions! The 'thing' consisted of three black metal bits and a little interface to fit in the expansion port of the QL. I managed to work out how everything went together - the QL sits in a compartment towards the front, a 64 way cable running underneath to the four expansion ports, which face away from the back of the QL. Over the top of the ports sits a large monitor cover, which is open at the back so that the numerous cables can trail all over the place. I quite like the box because it has a sort of wrist support ledge at the front to provide some comfort for my lazy typing style.

EXPANSION BOX PROBLEMS

Lacking an instruction manual, I wasn't quite certain that I had put everything together correctly. I rang another Quanta member who had purchased a Quest box, and was reassured that the 'thing' was supplied with the right number of parts. He was also able to point out where to plug in the extra power supply required for the four slots. This is near the front of the console, beneath a ventilation grill. This grill is perhaps one of its biggest design flaws, since a spilled drink would slop onto the power socket and circuit board. I think that this would make the Quest console unsuitable for general office use, but it is still worthwhile for a careful home user.

When testing the console, I used my Superdisk interface. It worked perfectly in the three slots which I number 1 to 3 from the expansion port side of the QL but would not work in the fourth slot. I resigned myself to losing the use of one port, and convinced myself that even a three slot board was good value at £40 since there was nothing else available which compared in design or price.

THE QUEST GOES ON

Quest had stopped advertising their QL products before I bought mine, and Medic and some other suppliers had also dropped out of the market before the Amstrad sellout for one reason or another. After my first two Quest purchases from different suppliers I was beginning to get the idea that a trickle of unused Quest items was emerging from a dusty storeroom. The announcement of PC Emulators for the QL made me think about purchasing a 5.25 inch disk drive, to make it easier to obtain PC software. When I saw an advertisement in Micro Mart for twin 5.25" drives and interface, I was not too surprised to find that they were of Quest origin.

The supplier did not have a QL at all, only a BBC and an Atari ST, poor soul. Therefore he was unable to test the interfaces he had, but had proved the drives to work with his BBC micro. His asking price seemed reasonable compared to the cost of buying new drives alone, and considerably less than the £700 or so Quest used to charge back in 1985. This time I was not surprised at the size of the box the drives arrived in!

GIANT DRIVES

Compared to the modern 3.5" single drive, the twin 5.25" unit seemed huge. I had not used this size disk since 1985 when I used an original IBM PC at work, so it seemed a return to the bad old days. The disk interface was also very big, housed in a black metal box similar to the Quest RAM. Finally, the driver software came on a single microdrive cartridge. No interface ROM at all - very old hat. The software adds the FDV device driver whereas most interfaces use FLP. It is necessary to have the software present in microdrive one to format a disk, which requires unusual syntax:-

e.g. `FORMAT FDV1_82_DISKNAME` for an 80 track double sided disk.

Quest supplied 180K, 360K and 720K drives, single or twin. My 720K drives can format to the lower capacities and read disks formatted on other QLs. The drives were made by Tandon, the original big 5.25" drive manufacturer, but are in a Quest-badged black box in a style which complements the QL.

My first attempts to use the drives ended in failure and frustration. I couldn't get the interface to work in the Quest expansion console, though the drives worked to a degree with the Sandy interface on the other QL. The supplier obliged by sending a spare interface to try out. In the end I got a working combination, but had to take the QL out of the expansion box, thereby losing the use of the extra 256K memory. I assumed that some sort of hardware incompatibility had caused the initial problems preventing the drives from working - just one of those things with the QL.

REDISCOVERING A QDOS RIVAL

The only review I had seen of the Quest drives referred to them as being supplied with the CP/M-68K operating system, utilities and C compiler. This wasn't supplied with the ones I purchased, but only a few weeks later I saw yet another advertisement in Micro Mart (I'll have to stop reading that magazine!) for CP/M-68K at £4. The advertiser had been given the software, which comes on one cartridge and four 5.25" disks, but had been unable to use it as his QL was broken. In fact, without a Quest disk interface or operating system module (a sort of expansion port dongle) he would have been unable to use it. I proved this by testing the software on my other (non-Quest equipped) QL - it locks up the QL after printing an appropriate error message. It worked correctly on the other QL, but I think this QL curiosity deserves full and separate mention.

Running CP/M-68K on the 128K QL seemed to produce a few 'Insufficient memory or bad file header' error messages. (How's that for a meaningful message!) I thought some files might have been corrupt, but thought that some way of adding extra memory again would be worth a try. The opportunity came when my wife went visiting and took our Bull Terrier with her. This gave me the chance to spread bits and pieces over the dining room table without the risk that the dog would leap on it and flatten a few bare circuit boards - a distinct possibility if he had been around.

THE MONSTER TAKES SHAPE

While trying to get the hang of CP/M-68K I had placed a few free advertisements in Micro Mart and other publications. One response, the only helpful one, was from the Quanta member who had sold me the expansion console. Eventually he was able to supply a user guide for CP/M-68K, but in the meantime he made some suggestions about the black box which set me experimenting again. With more space to spread out, I was able to try all the different permutations with the slots.

It proved that the Quest disk interface will work in the expansion console, despite it not having ROM based driver software, provided it is placed in slot one, i.e. the one nearest the QL expansion port. I also added the Sandy disk interface and parallel printer port. It is possible to add several disk interfaces to a QL if they use different device names. Quest used FDV, Sandy used FLP, so now I have FLP1_ (3.5") and FDV1_ & FDV2_ (5.25") on one QL.

Almost by chance, I tried the Quest 256K RAM in the supposedly non-working slot four of the console and..... it worked! For some reason slot four is not the same as the other three - it will only accept Quest expansion memory. I had previously tried it with the Sandy 512K with no success.

SOFTWARE/HARDWARE INCOMPATIBILITIES

Having assembled a 384K QL with single 3.5", twin 5.25" disk drives and parallel port, I thought I was in luck. I had one slot free and a SPEM QEPROM board, allowing three ROMs to be fitted, still to find a home for. I had already worked out that QFlash and Speedscreen ROMs cannot co-exist on the QEPROM board for some as yet undiscovered reason. I tried the Speedscreen ROM on its own on the board in slot 2, but it would not work, giving a corrupted message on the F1/F2 screen. I swapped it for QFlash, which did work, but caused further problems. The RAMdisk driver seems to prevent the Quest disk driver software, whether QDOS/FDV or CP/M-68K, from working. I am quite prepared to believe the Quest software is at fault in view of a number of other bugs I have discovered in it. Problems with the CP/M-68K software will be discussed elsewhere. A major flaw in the FDV driver is that there seems to be a 32K limit to the size of machine code files. This bug is apparent in both version 1.04 and 1.05 of the Quest software. I would be very grateful to hear from anyone who has a more up to date version of the Quest software, or who is capable of fixing the bug.

Because of the problems with the QEPROM board (or the ROMs), I have swapped it over to the other QL where it sticks a few inches out of the side with the excellent QFlash RAMdisk and toolkit. Consequently I still have a slot spare and ideas what to do with it, finances permitting. The Rebel Electronics hard disk interface would be an ideal purchase from my point of view, but I would need a bank loan to raise the funds. Alternatively, I would be interested in The Transformer PC Emulator on an expansion card. Schon are advertising this as forthcoming and I believe that the QL Super User Bureau are also going to be stocking it, when it's ready! This might interest those people put off by Schon's poor reputation for after-sales backup.

The number of products available to put into my spare slot seems to have diminished of late. I will keep scouring the pages of you-know-what magazine and Quanta. Devices I have heard of include QSound/QPrint from ABC Electronic, mono sound, stereo sound and speech boards from Maurice Computers, the eeprom programmer from QJump, and the QControl board from Care Electronics. I would like to see users' reviews of these products and details of any I might not have heard of (excluding fairly standard RAM expansions or disk interfaces).

THE STORY TO DATE

With the exception of the problem with the Quest software, my hardware setup is very satisfactory. I have spent far more on the QL than I had ever intended, but many of the purchases were at bargain prices so things would have been worse if paying full price. Some of the purchases have proved a challenge, but in a way this is the main purpose of the QL for me.

The position at present is as follows:

QL 1: upgraded to JS
 640K RAM
 Adman Services replacement power supply
 Toolkit II and QFlash on ROM
 Tandata QConnect modem set
 Epson LX800 printer
 Quickshot II joystick
 Ferguson TX tv/monitor

QL 2: JS ROM
 384k RAM
 2 Sinclair power supplies
 Toolkit II and ICE on ROM
 Sandy 3.5" disk
 Quest 5.25" disk (x 2)

The Serial 8056 printer is in retirement. I don't know whether it is possible to get these repaired cheaply, but the print head on mine seems to produce streaky output on the thermal paper.

These are networked successfully, but I am adding a serial cable connection to allow CP/M-68K to QDOS data transfer. I haven't yet been able to test the Quest drives with The Solution PC Emulator (as it is still 'in the post' I assume). DP mention problems with MCS disk interfaces in their latest advertisement. The Quest drives might also be a problem as I think The Solution relies on the direct sector access present on interfaces with QJump ROMs. If this is a problem, they should mention it more explicitly in their advertising to save disappointment. I will still be able to use it with the single Sandy drive. I hope it networks.....

As I am due to move to within 10 miles of Leyland later this year, I may be able to demonstrate the Quest stuff at a future Northern Sinclair Show should David Batty be encouraged to organise more. In the meantime, I'm hoping everything holds together.

Michael L Jackson

BITS AND PIECES

XLISP

I have supplied several people with a P.D. version of XLISP. However it has been pointed out that the cursor keys and delete functions do not work (probably due to incorrect mapping of PC keys onto the QL's (I'm guessing)). XLISP is written in C so if anyone cares to have a look at this with a view to fixing it - please get in touch. I also wrote to Ralf Biedermann (the author) to see if he had any solution to the problem but he reported he hadn't. There is now a V2.00 available for the ST if anyone would like to look at that - we can port progs. over from the ST quite easily (depending on the directory structure.) Ralf did, however, kindly send a disk of his own Public Domain software in archived form. He's written a prog that will compress progs etc together and uncompress them. (£3.30 plus a disk gets you this package.)(62 items! German texts)

INTERNATIONAL PUBLIC DOMAIN EXCHANGE

CGH Services have agreed to exchange P.D. software with Sinclair QL User Club in Germany and QL Contact, France. So, if you can translate German there's a couple more disks of software sitting here waiting for translation. We await the French collection with trepidation. Any other inter/national QL groups with substantial PD collections to exchange, we'd like to hear from you! N.B. Please don't use progs from magazines unless you have the copyright holder's permission or unless it says that the prog is P.D.

FRACTAL REPORT

This irregular journal (20 - 24 pages A4) is edited by John De Rivaz and published by Reeves Telecommunications Laboratories Ltd, West Towan House, Porthtowan, Cornwall, TR4 8AX and costs £10 for 6 issues. Each issue is packed with progs and articles on fractals, mandelbrots and julias. Some, at least, of the progs are in QL SuperBasic and others should be convertable. (Anyone like to write a guide to converting between different Basic dialects ?) I've sent John the mandelbrot progs from our P.D. library - but more are welcome. Rich Mellor has translated the screen prompts, so these are now usable.

TANDATA MODEM STACKS

Dave Fullerton has also had problems with his modem set - this time blowing some other chips out when attempting to use it. Has anyone else had problems of this sort with it? (Mine works O.K. Maybe Dave had a power overload problem?)

QPTR

Ian Bruntlett, QL Forum (no relation) 25 The Broadway, High Barnes, Sunderland, SR4 8LP has asked for anyone who has succeeded in using QPTR to contact him. He'll also distribute working examples of programs using QPTR. Ian also offered to review The Thing manager from Jochen Merz ages ago - so he should have a pretty good idea of how it works now.

SUPERTOOLKIT II

WRITTEN BY TONY TEBBY.

QJUMP / CARE ELECTRONICS

800 ST ALBANS ROAD, GARSTON, WATFORD, HERTS, WD2 6NL
AVAILABLE EITHER AS EPROM OR ON MDV_ .

I had better start this review by saying that this is not intended to be a highly detailed or technically informed article. It should rather be read as a beginner's guide, for non-programmers, to allow such people to evaluate whether there is any advantage for them in buying this toolkit. Neither is this a comparative review mainly because I did not have much to compare it with. (The toolkits and utilities available on P.D. whilst very cheap, do not compare as they have so few facilities.)

The first problem facing potential users of this toolkit is availability. Here there is no problem. Care got the Eprom version of this to me 18 hours after I phoned in an Access order. Excellent. The toolkit is also a component of many Disk interfaces. My JM QL uses a Sandy SuperQBoard which includes a Toolkit II, the Eprom is fitted to my JS QL.

The next consideration is ease of installation. As both versions I have are hardware rather than software, this is very easy. Simply typing "TK2_EXT" initialises the toolkit, after which it stays in the background until you use one of its keywords. The only problem I had with the Eprom version is that I cannot use both it and my Speedscreen Eprom at the same time. However as Speedscreen is used with finished progs and Toolkit II is used mainly with file handling, editing etc this isn't a big problem. As both progs are available on mdv as well, it would be quite possible, I imagine, to have both active at the same time.

Both the Eprom versions come with a small neatly printed manual. Although the printing is clear, I think the type may prove too small for some people - although it could easily be enlarged on a photocopier if necessary. Another point associated with the size of the manual is that it quite easy to overlook potentially useful commands. There is no index as such, but the manual has both a table of contents and an appendix giving a list of the keywords and which section they can be found in. The manual does give some quite clear examples, but space considerations have prevented examples being given for all the commands, and there is little guidance on how best to use commands, sample programs and so on, which limits its use for non-techies.

Activating individual sections of the toolkit is quite simple. The syntax is usually given in the manual and can either be incorporated into a SuperBasic program or used as stand alone commands. Thus if you are programming for your own use, you can write program incorporating Toolkit II commands, but if you want to distribute the program you'll need to be certain that the people at the other end also have access to the Toolkit too.

O.K. what uses can non-programmers put this toolkit too? Well the first use I discovered was to do with file maintenance. Anyone who needs to do a lot of copying of files from one media to another, as I do, knows how laborious this is if you try to copy each file individually. Some disk interfaces have Toolkit II as standard, but my Silicon Express one hasn't, and to copy files I needed to load a utilities disk and go through a slow menu-driven process. With Toolkit II, all that is a thing of the past, as a simple "WCOPY DEV1,DEV2_" gets things moving. Using the "Y/N/A/Q?" (Yes/No/All/Quit) prompt for each file means that the user can pick and choose which files get copied. Very useful. Another very useful feature is the ability to rename files. Under SuperBasic this is extremely tedious. Using "RENAME file1 TO file2" gets it done very quickly. There is also another command WREN which allows "wildcard" copying - essential if you're trying to load files or progs that have non-printable characters, such as u's with the dots over them, as found on some German progs.

SuperToolkit II makes a lot of use of default directories etc (shown by the command "DLIST" which saves some typing, but it can cause problems if you're not watching the on-screen prompts, as it will, for example, copy files to the default drive if it cannot copy to the drive you specified, with the file prefix of the attempted destination, which may lead you to think you've copied something that you haven't. On the other hand this may be a benefit in some circumstances.

One of the nice additions to the the standard QL operating environment that the toolkit brings is a simple editor, called ED. This can be used on progs written from scratch or going through loaded SuperBasic progs. Invoked by typing ED, you are then deposited at the head of the file where you can alter lines to your heart's content. However the facilities are extremely limited. I would have liked a simple "Search" facility to find, for example lines in a Basic prog starting "MISTAKE" ~ indicating that non-printable characters are being used.

Another facility that SuperToolkitII has is to improve Networking on QL's. Those of you with long memories will remember that one of the original selling-points of the QL was its ability to network, only the hardware on the early machines wasn't up to it. Indeed pre D-14 serial number machines may have totally non-functioning network ports. So it was with some trepidation I likned up a D-05 and a D-09 machine. Well, thanks to an article in QUANTA by Alex Tagg, I got them to work and simply by using the prefix "nl_" can copy files from one QL with a 3.5" disk drive to one with a 5.25" drive. And at some speed as well. I haven't done any timings yet but it looks quicker than mdv_ to mdv_ on the same machine and only marginally slower than disk to disk. So if you have two QL's ~ and they can be any memory size ~ it is well worth investigating networking. If nothing else you can use the second machine as a printer spooler. (Wehope to have an article on networking in the next issue.) My only criticism is that neither in the QL manual or the ToolkitII mamual is there a simple test program to see whether the network is functioning properly. (Computer Shopper have also been doing articles on networking recently ~ so check them out too.)

Other useful commands that "beginners" can use include: "STAT", "WSTAT" AND "ASTAT" which give useful statistics on the files; WDEL which allows one to step through the directory deleting unwanted files (using the Y/N/A/Q? prompt); HEX\$, BIN\$, BIN and HEX for converting between hex and binary; FREE_MEM which displays the amount of free memory; WMON and WTV which reset the windows to the default monitor and TV states. Now many more experienced users of the toolkit may be wondering why I have omitted so many of the other features. The reason is simple, I haven't worked out how to use them or they are not much use to me. Such features include: an ALTKEY command; cursor controls; error handling; clock; direct access files; jobs control, etc. Perhaps users more experienced in the use of Toolkit II will write in with some advice on these features.

Overall then, I can thoroughly recommend the SuperToolkitII. The range of functions available allows you to do things with your QL that were either impossible to do before, or at least were very awkward. Also, the fact that there are a range of features both for beginners and more advanced users means that everyone should find something of use in it, and have scope for learning more about the QL and SuperBasic as they grow in confidence with the Toolkit. As I said above, I'd like to publish more articles on using this, and other Toolkits as they are both very good value for money and educational tools as well as utilities that make life that bit easier for QL owners.

Note: I haven't discussed compatibility problems in this article. What I would recommend is that programmers make a note of the names of the functions etc used in this and other toolkits and avoid using the same names for their own procedures, variables etc. I would expect this toolkit to be compatible with QRAM as they come from the same stable, but beyond that I can't say.

Richard Alexander

1. Using Numbers in Machine Code and the Outside World.

Machine code books abound for home computers, and the QL is no exception, but the real difficulty is that not many books actually take the trouble to teach those who are brand new to machine code how to start.

So in this series of articles, I hope to be able to give everyone a little of my home grown knowledge about machine code programming.

A. What is machine code?

Computers cannot understand English, since they work wholly in numbers. This created many problems in interfacing with the outside world, and so BASIC was invented for mere mortals to use. The problem with BASIC is that it is slow, since it requires an 'interpreter' to convert each line into the numbers which the computer understands, before any action can be taken.

A much better answer to the problem in speed terms is to use machine code.

This is actually written by human beings in a language called assembly, which is made up of very short, mostly one-step instructions, called mnemonics. It looks very different from BASIC and yet to those well versed in machine code, it can be just as readable (or just as unreadable depending on the original programmer).

The assembly program is then converted into numbers for use by the computer by a program called an 'ASSEMBLER' which does the conversion process in one huge chunk. It is this which creates the greatest upturn in speed. A BASIC Compiler is very similar except the intricacies of BASIC mean that a compiler cannot get the same speed out of the processor as someone who wrote the program in assembly code in the first place.

B. Okay, so what now?

Well, a good place to start is getting to understand how the QL stores numbers. There are several ways of depicting numbers (different 'bases'), but whilst we humans use base 10 to represent numbers (or DECIMAL), unfortunately computers have to use base 2 (or BINARY). So you can see that some knowledge of binary must be useful in writing machine code. Also some knowledge of base 16 (HEXADECIMAL) is useful, since many debuggers use this base to represent numbers, believe it or not, "for simplicity's sake".

Do not worry if you are not too numerate, since base conversion is quite simple.

C. Number representation

To represent a number in any base, draw up a row of 8 columns (for example) across a page and label each column from right to left, 0 to 7. (The actual number of columns will alter depending on how large a number you are dealing with).

Next you take the base number (2, 10 or 16) and head each column by the base to the power of the column number.

For example, using decimal (base 10), and using '^' to represent 'to the power of':-

7		6		5		4		3		2		1		0		(column number)
10 ⁷		10 ⁶		10 ⁵		10 ⁴		10 ³		10 ²		10 ¹		10 ⁰		(column value)

REMEMBER anything to the power of zero equals one and anything to the power of one equals itself.

eg. $10^0=1$
 $2^0=1$

$10^1=10$
 $2^1=2$

Next, work out each column value by multiplying the base by itself (power number of times) -

eg. $10^3 = 10*10*10 = 1000$
 $2^5 = 2*2*2*2*2 = 32$

You would normally use a calculator to do this, although conversion by hand is in most cases unnecessary, since many QL toolkits contain commands to do it for you, for example BIN\$(decimal number) in TKII, will convert a decimal into binary.

Now you will have to fit your number into the table you have set out.

Base 10 is obvious since you are used to it:
for example the number 1432 fits into the following columns -

1 into 10^3 (or 1000 column)
4 into 10^2 (or 100 column)
3 into 10^1 (or 10 column)
2 into 10^0 (or 1 column)

This can be converted back by taking the number from the column and multiplying it by the column value ie:

$1*1000 + 4*100 + 3*10 + 2*1 = 1000 + 400 + 30 + 2 = 1432$

Other bases

Now the thing which puts people off using other bases is the appearance of the number, for example:

35 in binary = &1000011. In hexadecimal = \$23
(The '&' and '\$' symbols are widely used to emphasize that the number is in binary and hexadecimal respectively).

To convert back, place it into your column table and multiply out as you did for the decimal number.

If you do not follow this then please try and go through it all again and actually try bits out on paper, since this part is imperative to all machine code programming.

To convert into a given base from decimal, it is quite simple, you merely have to look for the largest column number closest to your given decimal number, then subtract this number from your decimal, adding one to the column. You then continue to do this until the whole decimal number has been converted.

For example to change 5121 to hexadecimal---
columns= $16^3(=4096)$ $16^2(=256)$ $16^1(=16)$ $16^0(=1)$

number	hex equivalent
5121	\$0000
1025 (5121-4096)	\$1000
769 (1025-256)	\$1100
513 (769-256)	\$1200
257 (513-256)	\$1300
1 (257-256)	\$1400
0 (1-1)	\$1401

To check....

$$\$1401 = 1*4096 + 4*256 + 1*1 = 4096 + 1024 + 1 = 5121(\text{decimal})$$

As a last point, you may have noticed that in hexadecimal, each column may have to hold a number greater than nine, since each column can contain up to 15. So, to keep things relatively simple, the numbers 10 to 15 in hex are represented by the letters A to F respectively. This means that each column may contain the following:

Binary 0,1
Decimal 0,1,2,3,4,5,6,7,8,9
Hexadecimal 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

D. So much for bases, how do I use these numbers now?

Well the QL can store three types of numbers. These are long words, words and bytes. These dictate the maximum value that a number can contain. The QL only operates in Binary, and therefore these descriptions relate to the length of a binary number.

A Long Word - this can be up to 32 columns (or bits) long and can therefore contain numbers up to $(2^{32})-1$.

A Word - this can be up to 16 columns (or bits) long and can therefore contain numbers up to $(2^{16})-1 = 65535$.

A Byte - this can be up to 8 columns (or bits) long and can therefore contain numbers up to $(2^8)-1 = 255$

The different lengths and the ability to manipulate binary numbers are very important to QL machine code, since these can alter the speed at which a program operates and also the limits upon a program.

2. Writing a Program in Machine Code

First of all, it is important that anyone who is going to write a machine code program has access to a good editor (I can recommend the ones by Ark and Digital Precision) and also a good assembler. I have only tried the Metacomco one, and although it is now rather old and still contains a few minor bugs, I can recommend it. I'm also told that the one by Talent is also excellent and includes a disassembler/monitor - essential for sorting out problems in other peoples' code.

So how do you actually begin writing a program?

First of all, KEEP IT SIMPLE! This important rule will ensure that any bugs can be found pretty easily, and if you write your program in small sections, each of which can be tested on their own, you will find it much easier to follow what is happening and correct anything which causes problems.

By way of an example, I shall proceed by writing a machine code program to do the relatively simple task of clearing the whole screen, and printing the message 'CGH Services For Adventuring Needs' at 10,15.

In BASIC this would be quite simple, just use the lines:

```
10 WINDOW 512,256,0,0:PAPER 0:CLS
20 INK 7:AT 10,15
30 PRINT 'CGH Services For Adventuring Needs'
```

Before we can begin to convert this to machine code, the whole task must be broken down into its component parts, since Assembly language can only understand very simple one step instructions.

So, the program can be broken into the following:

- a) Change the WINDOW to fill the whole screen
- b) Change the PAPER colour to 0 (black)
- c) Clear the whole screen
- d) Change the INK colour to 7 (white)
- e) Move the text cursor to the position 10,15
- f) Print the message

Now, before even starting to think about converting this to machine code, there is a very important step. For this you will need a good book on QDOS (the QL operating system). I can recommend Andrew Pennell's 'QDOS Companion' and Colin Opie's 'Assembly Language Programming'.

Looking through a book such as this you will see that QDOS contains lots of useful little routines to help you write your program, which means that most of the work is already done.

Routines which would be useful here are:

IO.OPEN (Trap #2) = Open a channel
SD.SETPA (Trap #3) = Set the paper colour
SD.CLEAR (Trap #3) = Clear a window
SD.SETIN (Trap #3) = Set the ink colour
SD.POS (Trap #3) = Set the cursor position
IO.SSTRG (Trap #3) = Send a string of bytes to a channel.

(The description on the left is the name of the routine, and the Trap number is part of a machine code instruction as to how we actually call the routine).

So, now for the finished routine:

* Print Out Simple Message

*

* Rich Mellor 1989

*

* Define some (variables) to use in the program:

* Trap #2 instructions

			COMMENTS
IO.OPEN	EQU	\$1	(LET IO.OPEN = \$1)

* Trap #3 instructions

SD.SETPA	EQU	\$27
SD.CLEAR	EQU	\$20
SD.SETIN	EQU	\$29
SD.POS	EQU	\$10
IO.SSTRG	EQU	\$7

* Now for the actual program

START

MOVEQ	#-1,D1	Load D1 with -1 (like LET D1=-1)
MOVEQ	#2,D4	
LEA.L	CHAN,A0	Point A0 to the channel name
MOVEQ	#IO.OPEN,D0	
TRAP	#2	Call QDOS routine to open channel
MOVEQ	#0,D1	
MOVEQ	#-1,D3	

```

MOVEQ    #SD.SETPA,DO
TRAP     #3
MOVEQ    #SD.CLEAR,DO
TRAP     #3
MOVEQ    #7,D1
MOVEQ    #SD.SETIN,DO
TRAP     #3
MOVEQ    #15,D1
MOVEQ    #10,D2
MOVEQ    #SD.POS,DO
TRAP     #3
MOVEQ    #34,D2
LEA.L   MESSAGE,A1
MOVEQ    #IO.SSTRG,DO
TRAP     #3
MOVEQ    #0,DO
RTS
MESSAGE  DC.B    'CGH Services For Adventuring Needs'
CHAN     DC.W    15
         DC.B    'CON_512x256a0x0'
END

```

Note that the last line ('END') must appear at the end of any machine code program to tell the Assembler where the end of the program is.

As you can see, the program in machine code looks quite long, and is in fact a pretty inefficient way of carrying out such a simple task. However once it has been run through an assembler, it will in fact occupy less memory space than the equivalent BASIC program.

Why do some lines begin with a '*'?

These lines are similar to REM statements in BASIC and will be ignored by an assembler. They are merely there to help make sense of the program.

Why is it in columns?

Each column means something to an Assembler.

The first (left) column, is merely a label, equivalent to line numbers in BASIC.

The second column contains the actual machine code command.

The third column contains the variables and numbers used by the machine code command.

The fourth column is unused by Assemblers and can therefore contain anything you desire - normally a comment on the line itself.

So what exactly is the machine code doing?

First of all, some variables are defined to make the program easier to understand. These are carried out by the first six actual program lines, by the command EQU. Assembly language on the QL is restricted in the number of variables you can use. You can use as many of these EQU lines as necessary, but they merely replace a number by a word (or label) and cannot actually be used within the program to store anything.

In fact, a machine code program can only use the variables D0 - D7 and A0 - A7 to carry out any operations. NOTE that A6 and A7 have a VERY restricted use and are not available as ordinary variables.

At the label START, the program proceeds to open a channel using the QDOS routine IO.OPEN. To do this, the variable D1 must contain a number telling QDOS who it is that

is calling it (-1 means it is the current program, or job). Next D4 is loaded with an 'CHANNEL ACCESS CODE' which in this instance has no real effect, but merely tells QDOS that it is a brand new channel.

Next, QDOS must be told what the channel is to be, using the normal channel name as used by BASIC, ie 'CON_512x256a0x0'. This name is held at the label 'CHAN' and is preceded by a WORD telling QDOS the length of the string.

If you look to the label CHAN, you will find that the program uses the commands 'DC.W' and 'DC.B' before the actual data of the name. These commands are equivalent to the BASIC command 'DATA'. They are both the same, except the '.W' or '.B' afterwards tells the QL that the data is in a word or a byte respectively.

To tell QDOS where the channel name is stored, the command LEA.L is used, which merely points A0 to the label (or makes A0 equal the address of the channel name). The QDOS companion informs you that A0 is the variable needed by QDOS to point to the channel name.

Next, as per normal QDOS syntax, D0 must be loaded with the address of the routine, which happens to be the number 'IO.OPEN'. Then the routine is actually called by the command TRAP #2

This routine will return the channel id number in A0 which is required by most of the other QDOS routines pertaining to the display. This channel id number is related to the BASIC '#channel', in that it is used to identify which channel is to be used by the routine. However, the number itself is not as easy to understand as the BASIC equivalent, since it is returned as a BINARY number which represents both the channel number and the number of channels which have been previously opened. For example the BASIC channels #1 and #2 have channel id numbers of &00010001 and 00020002 respectively. &00010001 means that it is channel #1, and one channel has been opened previously. If you were then to open channel #4 from BASIC (you cannot choose the channel number in machine code), this would have the channel id &00040003.

The next routine to be called is SD.SETPA, which sets the PAPER colour. On entry to this routine, QDOS expects the new paper colour to be held in D1 (which we do by the command MOVEQ #0,D1), and the 'timeout' to be held in D3.

The 'Timeout' is actually the time which you have allocated QDOS to perform the task. A setting of -1 ensures that your program is not returned to until QDOS has finished.

Similar operations must then be performed for the other calls. The main difference is the passing of certain parameters, such as with SD.SETIN, D1 must contain the new ink colour, and with SD.POS, D1 and D2 contain the new x and y coordinates respectively.

The routine IO.SSTRG is slightly more complex, in that here, the address of the string must be passed to the routine, by pointing A1 to the start location (MESSAGE), using the command LEA.L again. D2 must then contain the length of the string and finally the routine is called.

For further information on what parameters must be passed to each routine, see QDOS companion.

Finally, D0 is loaded with zero to signify to BASIC that there is no error, and then the machine code is stopped by the command RTS, which is similar to STOP in this example.

Once the above program has been run through your Assembler, it can finally be run by the commands:

```
a=RESPR(104):LBYTES mdvl_test,a:CALL a
```

I hope that this has served as a useful introduction to machine code programming. You could for instance alter the above program so that a different character size was used, or the text appeared on a different paper background.

The necessary QDOS routine to change character size is:

SD.SETSZ (Trap #3 with D0=\$2D)

On entry the following must be set:

D3=timeout

A0=channel ID

D1=character width:

0: 6 pixels

1: 8 pixels

2: 12 pixels

3: 16 pixels

D2=character height:

0: 10 pixels

1: 20 pixels

Rich Mellor 1989

BITS AND PIECES

QL CRASHING: How to avoid it

A few hints from Rich Mellor & Leon Heller

a) Swap the order of your plugs - it is possible that one socket produces more mains spikes than others.

b) Fit a 5 volt 2 amp regulator from Adman Services, 53 Gilpin Road, Admaston, Telford, Shrops TF5 0BG. Tel: 0952-255895. Cost 80p & p+p.

c) Try the QPOWER regulator from POWER Computing (now available from Care Electronics). Dave Fullerton, 45 South Street, Southville, Bedminster, Bristol, BS3 3AU, reported that a QPOWER unit zapped the ULA on his #2 mdv, but I presume this happened with one of the earlier versions, as later ones come properly screened. Anyone else had this problem ?

d) Check all your connections, leads, even the ones inside your plugs - maybe something is loose. (Recommended to editors!)

e) Leon Heller recommends keeping the feet on your QL as this improves the airflow, which should keep it cooler. (That's the QL's feet, not your own.)

f) Leon Heller, amongst others also reminds us that disks are not immune to the magnetic fields surrounding monitors and TV's. Consequently keeping disks on them is asking for corrupted data.

PSION PROGS

Rich Mellor has added his configure prog to the P.D. library to help set up the Psion Progs (Quill etc) as you like them but he has not been able to sort out where the DEFAULT page setting, etc is in the program. If anyone can inform him then he can add the ability to set up the DESIGN options as you'd like them on loading the Psion progs (instead of having to load templates).

ASSEMBLER/DISASSEMBLERS

Jean-Yves Rouffiac, author of Dreamlands, has written in asking if I can recommend an assembler/disassembler for the QL. I suspect this is best answered by people who use the things - so over to you lot. Jean-Yves can be contacted at 31 Hampstead Hill Gardens, London NW3 2PJ.

SPARE PARTS AND REPAIRS

Joe Atkinson has access to a wide range of spare parts for the QL at very reasonable prices. Contact him at 36 Ranelagh Road, Ealing, London W5 5RJ.

One of my QLAF correspondents has reported having appalling trouble getting his QL repaired. Apparently what started life as a JS QL came back, after several months, as an AH with two keys the same. It took the intervention of QLWORLD's trouble shooter to get even that much out of the repairers. So recommendations please for efficient repairers of QL's.

BILLBOARD BAGGINZ SWALLOWS A FLY

REVIEW OF DIGITAL PRECISION'S IDIS DISASSEMBLER, £24.95.

I was walking past Waz Lane, seeing the last remains of the Orc Massacre Inn being demolished, when suddenly this big square fly dive bombed me gob.

(Cough) I said. Well what would you say, Mr. Mega Lungs 1989? Anyway, you'll never believe this, but can you guess what that cough brought up? The fly! Except it wasn't a fly! That's right, a fly that wasn't a fly. Well I said to myself, "Billboard, something very strange is going on here!" It was one of those "portable data-holes" which those QL users go freaky about when they discover that they have accidentally "formatted" the wrong one. And it had some arcane runes on - I'll try and write them down here - "IDIS DISASSEMBLER". It probably meant something very obscene in Goblin, but I thought I'd take it along to Lirpa down at BritMat U.K.

Lirpa wasn't too happy. He was ranting and raging about the enormous cost of (fresh) Gorgon's heads. And the drop in demand for leeching - not surprised, it's a dreadful habit. Anyway, he perked up a bit when I showed him this square fly. "Ah an eye diss emm dee vee" he crooned (funny people these Necromancers.)

He placed it in a bigger square fly, one he called "Kyou Ell". Well lights flashed, the "emm dee vee" buzzed (well there's one normal thing that happened today - at least square flies can buzz like normal ones, I said to myself). Lirpa merely grunted the cryptic remark "Shall we incorporate it in my Boot file?" Well if he wanted to have a partner in his foot wear portfolio, that was alright by me.

I asked Lirpa what he was doing with the square fly. He was preparing to teleport in something. It was a bit early in the morning for incantations, but I said nothing. I hoped it would be something useful like a couple of Pegasi. Huh, I should have known better. Four leaves of runes on pea green paper from a place Lirpa called "Dee Pee". So that's how they/it got their name. Lirpa read the runes and pronounced it to be the "man ewe arl", probably Necromancer's jargon for "manual".

I could tell Lirpa was a bit disappointed by the manual. To quote, "Blah! two leaves of text out of four leaves. Blah! No index", (I'll skip the rest of the Blah's, it was like listening to [I'd better be careful wot I rite, in case I get turned into a frog] someone who was trying to learn "Baa Baa Black Sheep"). Then he started pushing his "Kyou Ell" so that his magic picture frame would show different pictures. He called it SuperBasic. I called it weird (I was living dangerously, through boredom).

Lirpa said he was installing "Eye Diss". He wasn't too happy with the "numberless" boot, saying he had to "type EDIT, delete the line number and type MERGE ram1 Edit" to get a comfortable look at the "listing". Lirpa got a bit technical: "Dunno why there wasn't more installation info in the manual IDIS1 and IDIS2 are merely bits of SuperToolkit Two and Turbo Toolkit they can be omitted from MY boot but I'll nab the extension command WHERE what a dumb name for a proc or function lister and I'll nab ISIT may junk it later so update my Boot, copy IDIS_EXE and UTIL_EXE to floppy wan to see how it goes blast out of space on eff ell pee won soon fix that". This was all gobble de gook to me.

Then Lirpa got motivated, said, "Stuff that, wotzit like??". "Well how should I know", I said, so Lirpa, highly irritated, turned my nose green. So I listened.

"EX idis_exe"

A green box (same shade as my nose) appeared on the screen.

"IDIS V2.00 DIGITAL PRECISION (c) by PL 1987"

"ASCII / DIS / IDIS / QUIT :"

Lirpa said "I'll try askee furst".

So he pressed "A" and the green box said: "Output"

Pressing ENTER made IDIS write "screen" by itself, how jolly clever I thought, is this

the intelligence that Lirpa was crooning about?

"Start (& for decimal)?"

Lirpa seemed happy that the "numba bayse de faulted to bayse sick-teen". Oh hex, what an accent.

"\$0"

"Not a hex number"

"Oh yeah" said Lirpa.

"0"

Then "IDIS" showed lots of funny numbers. A green on black column of addresses (with funny street names like 000E10, how silly, I ask you) followed by three columns of hex long words and a black on green "representation in askee". Well I won't bother trying to make anything out of politics in askee other than like politics anywhere, it's very strange to an outsider.

Lirpa then sent the output to a "file" called "raml_test" while showing it on the screen. "Jolly clever", I said.

Lirpa wasn't amused. He said use Qmon to:

"o3 raml_test, d3 0 `numbytes", o3"

"Well what's so good about that?", I asked.

So Lirpa got on his hobby horse. "In Qmon you have commands and can define a command for flexibility and to save on typing. Dee Pee have written this in Soupa Basic but Turbo-charged it, which really naffed me off. Now I have a prog that I can't alter, because I don't have the source. Bang goes any filter production line linkages, and it isn't re-entrant. And there's no explanation in my manual (he said it properly this time) about UTIL_EXE and the intelligent disassembly should have been better explained. (Perhaps Billboard will brave Lirpa's wrath to ask him to explain what all that is in plain English, Ed.)

"Right, on to normal disassembly"

"D"

Output? Screen (i.e. enter)

Start (& for decimal)? etc...."

"Normal disassembly shows addresses, full hex opcodes and assembly equivalent" said Lirpa.

"Right, on to intelligent disassembly", he said, a bit more interested.

"I"

Output? raml_Test

Start (& for decimal)? etc "

3 letters for libel (A-Z)?"

What was that? Libel? Magic cross- words? Necromancy is being cheapened. Oh, I misread it. LABELS.

"3 letters for label (A-Z) ? TST"

Lirpa warbled on a bit: "Output from IDIS is surprisingly pleasant to follow, with labels being automatically entered instead of addresses. A pity that there is no option to include addresses or hex opcodes in the listing. All that has to be done now is to read it in an editor I'll try MCC ED "

"Was it worth it?", I asked.

"I'd prefer the source code to customise but, yes it should be evry useful. There really should be more intelligence. Here's five gold pieces - scam!"

Very touchy these Necromancers, so I sped off to the nearest inn to spend my gains.

Billboard Bagginz.

WILL SPARE PARTS SHORTAGES BE THE DEATH OF THE QL?

I have enough applications software, utilities and programming languages for the QL to keep me occupied for the next few years. It will take me long enough to master all the QDOS software, but the ability to use MSDOS programs on the QL will help to guarantee it years more use. However I am concerned that the QL may become unrepairable long before it has lost its usefulness.

The feeling of unease is due to Amstrad's copyright over the Sinclair custom chips inside the QL. I understand that supplies of some components may become difficult to obtain over the years and there is no guarantee that Amstrad will manufacture any more - or allow anyone else to do so.

There doesn't seem to be any problem at present. QL repairs are advertised regularly in Sinclair QL World, and some other repair firms advertise in Micro Mart, offering QL repairs from around £25. In addition, the QL Super User Bureau seems to offer the longest guarantee on repaired QLs, and very cheap spare parts can be purchased from Quanta. If you are prepared to have a go yourself, some of the most common faults, such as keyboard problems, can be solved for as little as £4. There is no excuse for letting your QL remain unrepared in a cupboard or attic!

While the QL is capable of being healed at the present, a pointer to the future is perhaps Commodore's attitude to the CBM +4. For those who don't remember it, this was Commodore's big rival to the Sinclair QL (before it tried the Commodore 128 and finally the Amiga). The +4 had 64K RAM and the four usual business applications built-in (on a 32K ROM?). It sold for about £350, but lost hands down to the QL in all respects other than price, and was still poorer value for money. Even before the official demise of the QL, +4s were selling for about £60 new. There is a tendency for the +4 to suffer failure of the 8501 I/C, as reported in Scotland's The Sunday Post. This paper reported the sad tale of a ten year old owner of a second hand +4 who had to wait over 14 months before a supply of the chip was manufactured. As the production run of 4000 chips was snapped up by dealers as soon as available this implies that there must have been thousands of useless Commodore computers up and down the country waiting to be repaired. I sincerely hope that such a position will not arise in respect of the QL within the next five years.

As far as I know, the Commodore +4 was compatible only with the CBM 16, a 16K sort of "Son of VIC 20" - barely a rival for a 16K Spectrum. At least there is the prospect of an upgrade path for QL owners requiring new hardware. Many people will be reluctant to sacrifice their QL-specific computing skills and convert to MSDOS machines simply to get more reliable hardware. The CST/Dansoft Thor and the ABC Elektronik Enigma would help prevent such a change of loyalties if they became available in decent quantities and at a realistic price. In the meantime, the Atari ST with QL emulator offers a good compromise, but lacks QL networking.

These upgrade computers still tend to mean that any QL hardware add-ons cannot be used, which is why I would prefer to see a new computer which is 100% hardware and software compatible with the QL. The interfaces and other expansions remain a problem area as very few repair firms seem capable of mending them, and some add-ons required custom chips available only from the original supplier - who might have quit the QL scene. I would be very surprised if any QL repair firm would know what to do with a Quest disk interface for instance.

It may be that the hard-core of 'QL Loyalists' will have to rely on obtaining replacement second hand QLs from owners deserting the machine for this year's latest craze (which may prove to be less sophisticated than the computer they are selling).

Michael L Jackson

QUEST'S CP/M-68K - A RIVAL FOR QDOS?

BY MICHAEL L JACKSON

CP/M on the Sinclair QL? That's a ridiculous suggestion, isn't it? Well, Sandy PCP didn't think so when they launched the first CP/Mulator, and neither did Digital Precision, who produce the Success emulator. These both allow Z80 CP/M programs to run on the QL by emulating CP/M 2.2. In addition, DP's The Solution PC emulator is said to allow CP/M 86, written for 8086-based machines, to be used on our own 68008 machine. But these are quite different from a version of CP/M actually written to run on computers based around the Motorola 68000 family of chips.

THE QL AND ITS COUSINS

It would be surprising to find CP/M on 68000 computers as it is an operating system with its period of success behind it - in the late 1970's and very early 1980's - whereas the 68000 machines began to enjoy commercial success only in the mid-1980's. In 1984 when the QL was born, the edition of Personal Computer News which reviewed Sinclair's new product mentions only the Apple Lisa, Corvus Concept, Fortune 32:16 System 2, HP Series 200 Model 16A, Sage II and IV, and the Tandy TRS-80 Model 16 as using the 68000 chip, out of dozens of machines using Z80, 8088/6 and 6502 chips. (There are a lot of catchy and memorable names in this list, aren't there?) The QL was the first computer to use the 68008, and its nearest cousins based on Motorola chips started at £3200, with most being more than £4000. In other words, the position was quite different in 1984 compared to today, as 68000 computers were very much in the minority so couldn't have presented much of a market for the developers of CP/M. Even now, the Macintosh, ST and Amiga have not overtaken (in terms of sales volume) the machines based on Intel 80x86 chips which run under MSDOS.

CP/M-68K - A PHOENIX FROM THE ASHES?

Despite the limited popularity of 68000 machines, a version of CP/M was written for them in 1983, presumably on the assumption that users upgrading to the more powerful computers would still like to be able to use an operating system with which they had become familiar. In 1985 Quest International Computer Technology Limited applied the same logic to the QL and launched its own version of CP/M-68K. A little later, Atari launched the ST with a disk-based operating system modelled on CP/M-68K - early TOS.

Quest supplied CP/M-68K in two versions - one at £60 for those who had added Quest disk drive interfaces to their QLs (the interfaces contain some of the system code), and one at £100 for microdrive-only QLs, with an operating system module to fit in the expansion port. Sinclair User said of the Quest drive system "If you want to run CP/M-68K then this is the only system around."

WHY BOTHER?

Sinclair User did not investigate the matter of why anyone would want to run CP/M-68K on a QL. As mentioned above, unlike the CP/M 2.2 and PC emulators available now, CP/M-68K would give compatibility with only a small number of other computers. Perhaps there was a certain lack of faith in QDOS? Admittedly there had been problems with the QL's own operating system from the start. At first it was going to be GST's 68K OS, which was slow and had a 32K size limit on basic programs. Then QDOS wouldn't fit in the 32K ROM space so there was a 16K kludge. And then the bugs..... But you all know that story.

However, in making comparisons with CP/M-68K I will define QDOS as being a JS ROM plus QJump's Supertoolkit II 2.10 or greater, since these were available by 1985 or thereabouts. I believe QDOS to be incomplete without TK2, but to add QRAM into the equation would make the contest a Mike Tyson vs. Larry Grayson battle.

THE QUEST BEGINS

I used to buy Sinclair User for its ZX81 coverage, so had read of CP/M-68K long before I owned my first QL. However, I had completely forgotten about it until late 1988 when the QL Super User Bureau asked its readers for information about it. Around that time I had bought the Quest twin 5.25" disk drive system, which prompted me to dig out the old magazine review. This told me I could run the 'antique' operating system, so a few weeks later when I saw an advertisement in Micro Mart for CP/M-68K I rang up the advertiser. The deal made, I waited eagerly for my £4.00 bargain.

Without the Quest manual I had some difficulties. A friend was familiar with CP/M 86 on a PC and his advice over the 'phone enabled me to make backup copies of the four disks. I had other problems, but most were solved when I increased the memory of the QL to 384K, which allowed the larger programs to run.

WHAT DO YOU GET FOR £100 / £60 / £4?

As the microdrive-only version uses a dongle in the expansion port, I assume you get fewer features for your extra £40, i.e. you cannot use external memory and consequently cannot use some of the utilities.

To use CP/M-68K, the disk operating software is loaded from a single unprotected microdrive cartridge, which turns off QDOS. The devices available depend on the particular QL, but on mine are:~

A: 100K MICRODRIVE
B: 100K MICRODRIVE
C: 800K 5.25" DISK
D: 800K 5.25" DISK
E: NOT AVAILABLE ~ RESERVED FOR EXTRA DRIVE
F: RAMDISK, SAY 100K

The four Quest disks contain:

- a) CP/M-68K transient utility programs;
- b) AS68, a 68000 assembler and associated utilities; and
- c) a C compiler and utilities.

To get to know the system I had to rely on Rodney Zaks' "The CP/M Handbook", which is essentially about Z80 CP/M 2.2. Surprisingly many of the commands worked in the same way, so I had enough to allow me to experiment a little. I also borrowed the manual for an Amstrad PCW 8512, which uses CP/M Plus, but found that there were different commands and the syntax was not the same.

PLEAS FOR HELP

In order to get more information about the software I placed advertisements asking for help, which led to me obtaining a spare copy of Quest's "CP/M-68K Operating System User's Guide". Unfortunately I still need the manuals for the assembler and C compiler, so if anyone has access to a copy, please let me know c/o CGH Services.

As is quite typical on the Sinclair scene, the manual describes programs which are not present on the disks and attributes features to other programs which they do not possess. The manual is in different typefaces, suggesting that in part it was a standard Digital Research manual, the rest being QL-specific material. I don't suppose Quest ever had much incentive to correct the manual because CP/M-68K doesn't seem to have sold well. I now have two copies of the disks, one dated June 1985 and the other dated September. There is a difference of only 30 in the serial

numbers on the labels, implying that this represents the number of copies sold in three months. I would suggest that a reasonable scoreline for a QDOS vs. CP/M-68K match would be 150,000 vs. 150 (sales).

OPERATING SYSTEM FEATURES

As I've not been able to get to grips with the Quest assembler or compiler yet, it wouldn't be fair to judge them against comparable QDOS products. Also CP/M-68K lacks a basic interpreter, so there isn't anything to compare with SuperBASIC (is there anywhere?). Comparisons are based on how devices and files are handled, plus other similar utilities.

Once the software has been loaded from cartridge, the following built-in commands are available:

DIR; DIRS; ERA; REN; SUBMIT; TYPE; and USER.

The QDOS equivalents are:

DIR & WDIR; DELETE & WDEL; RENAME; DO; and VIEW. The replacement for different user areas in QDOS would be directory navigation, using the following: DATA_USE, PROG_USE, DEST_USE, DDOWN, DUP, DNEXT, and DLIST. QDOS offers greater flexibility than CP/M user areas.

CP/M allows the use of wildcards in file name selection, e.g. DIR D:*.68K would give a list of all files ending in 68K on drive D. The QDOS equivalent would be WDIR FDV2__68K.

TRANSIENT COMMANDS ~ OS EXTRAS

As we all know, it is very easy to add extra commands to QDOS by purchasing a machine code toolkit or writing your own. Extra commands can be added to CP/M-68K in the form of transient utilities. Apart from those specific to the assembler or compiler, the ones supplied are:

BAUD; ED; FORMAT; MDCOPY; MDFORMAT; PIP; RAMDSK; and STAT.

To use one of these commands, for example PIP, type at the prompt "

C:PIP ^ENTER"

and a program called PIP.68K will be loaded from the disk in drive C. PIP stands for Peripheral Interchange Program, and its nearest equivalents in QDOS would be WCOPY, COPY_0, COPY_N and COPY_H. For instance, to copy all files from one disk to another:

CP/M-68K:~ PIP D:=C:*.*

is the same as

QDOS:~ WCOPY FDV1_ TO FDV2_

but doesn't offer the Y/N/A/Q? flexibility of WCOPY, and automatically overwrites all files with the same name if they already exist on the destination disk. This could cause problems!

ED is a line editor to create or amend text files. I find it difficult to use compared to a full screen editor such as QL Editor, but with the recent acquisition of a manual I am now having some success. FORMAT is used to format disks, MDFORMAT to format microdrive cartridges. MDCOPY is a potentially useful utility which copies files from QDOS microdrive cartridges to CP/M-68K cartridges or disks. It is only a one way transfer, but should allow assembler or compiler

source files to be prepared using a decent full screen editor (even Quill), then to be transferred to CP/M.

The STAT commands gives several different options to obtain statistics about files and devices, e.g.

STAT DEV: ; STAT VAL: ; STAT DSK: ; STAT *.68K ; or just STAT.

To obtain similar information, QDOS would require combinations of STAT; WSTAT; DATAD\$; PROGD\$; and DESTD\$. However, with QDOS it is possible to obtain information about memory usage, e.g. PRINT FREE_MEM, and file size is reported to a greater degree of accuracy and information about the date and time of file updating can be displayed.

The main advantage of CP/M-68K is that it is a 'standard' operating system, running on computers other than the QL. (However, I don't know which 68000 computers did use CP/M.) Anyone familiar with CP/M 2.2 would probably find it easy to use. Programs written on other machines should run on the QL. But here is the main drawback. I have been unable to discover any supplier of applications software for CP/M-68K, including two public domain libraries. At the best, I have been able to add two more utility commands - ERAQ68 and COMP68 - but there is no Wordstar or Supercalc (or near clones).

THE VERDICT

There are not many QL programs compared to the total for MSDOS machines, but they far outnumber the available programs for CP/M-68K. Any operating system without applications software is bound for extinction.

CP/M-68K is loaded from cartridge and takes up valuable memory space, unlike ROM-based QDOS. It seems slower than QDOS, has fewer commands, and is supplied with a C compiler rather than a basic interpreter, so will appeal to a minority of would-be programmers. It crashes with little provocation, for instance when a disk's directory is full (with 400K free space left on the disk!), or when the wrong key is pressed at the wrong time. It will only run on QLs with Quest hardware added.

I am not writing off CP/M-68K completely at the present. I have had more than £4 worth of amusement out of it and there is still the prospect of getting to use the assembler and compiler. If I can obtain the appropriate manuals, I would probably try to convert programs for this operating system. Perhaps PDQL's Superbasic-C-Port would allow me to translate existing SuperBASIC programs, as it produces C source code from basic input.

If you haven't got the appropriate Quest hardware, I reject Sinclair User's 1985 verdict and suggest that it is not worth buying in order to run CP/M-68K. Stick with QDOS. On the other hand, if you've already got the hardware, as I had, and can get the programs very cheaply you might get some fun or alternative experiences out of them.

Michael L Jackson

If any of our readers can help Michael locate any software that runs under CP/M-68 do let us know. The Public Domain Software Library at Winscombe House, Beacon Road, Crowborough, Sussex, TN6 1UL have the following CP/M libraries: CPMBBUK (111 disks); KUG (60 disks); SIGM (294 disks); UK (53 disks); Apple (56 disks); BOOG (59 disks); US National (93 disks); C Users Group (54 disks); Netherlands (33 disks) and hope to have another 50 disks available soon. How many work with CP/M-68 is another matter!!

Richard

LIGHTNING Digital Precision (£29.95)
 SPEEDSCREEN Creative Codeworks (£20)

Both of these packages offer something essential for QL adventurer's: an increase in speed of operation of programs coupled with relatively low memory consumption (see later). LIGHTNING is very similar to SPEEDSCREEN in that both programs alter the QL's display handler to speed up output, except in two very important aspects: LIGHTNING also speeds up graphics and mathematical functions - areas which SPEEDSCREEN left well alone.

Setting up both programs is relatively easy, since both have a configuration program which puts only a few key presses between starting up the package and having the new display driver invoked. Once invoked, the difference in speed between the two packages becomes apparent, especially if you are using the PSION programs, since you are able with LIGHTNING to switch off the resetting of all of the QL's windows whenever a MODE command is issued. This only helps to speed up programs which use the MODE command to tidy up the screen display (such as the PSION programs when F2 is pressed), since it may cause disruption to the screen.

The speed of actual text output is not very much different between the two - I include some comparisons, using the SPEEDSCREEN DEMO PROGRAM after compilation with TURBO (Digital Precision)(a). However, LIGHTNING also speeds up text output in MODE 8 and ALL paper and ink colours, unlike SPEEDSCREEN which does not affect this at all.

I have had no problems with compatibility with my other programs (b), especially since LIGHTNING does not encounter problems with programs which alter the MODE without resetting all of the INK and PAPER colours (such as MORTVILLE MANOR PYRAMIDE), unlike SPEEDSCREEN which throws up this problem quite often unless you have an alternative display driver such as that provided by QRAM (QJUMP) or T. Tebby's MG Rom patch. If a program does give problems when used with LIGHTNING, QRAM users will be pleased to know that they are able to turn off LIGHTNING even though QRAM is present in memory - something that you cannot do with SPEEDSCREEN since QRAM turns SPEEDSCREEN back on again!!

LIGHTNING also re-writes many of the QL's mathematical functions to make them much faster, allowing you to select the degree of accuracy using the command lngPREC. This command is most useful, since the lower the degree of accuracy needed by a program, the faster the mathematical functions are able to work. This may take a little experimentation to find the best degree of accuracy for each of your programs, but does not take too long since there are only three settings for the degree of precision. Note that you may not be aware that some of your programs (like the PSION set) use these mathematical functions, and benefit from the lowest degree of precision. - It would be useful if DP would release a list of the optimum settings for their programs (and maybe some of the more popular programs).

The graphics package makes quite a difference in nearly all of your programs, with all routines speeded up by between 50% and 600%. There is no loss of accuracy here, and even some of the QL's bugs are fixed, such as parts of circles off the screen are now printed correctly and the FILL command does not miss out any pixels within a shape. For example, the following command will now produce part of a circle on the screen -

CIRCLE -500,50,550

DP say that they have been careful not to fix too many bugs in case programs are written which rely on these 'features'. If you come across a program which does in fact rely on these features, the graphics enhancement package, like any of the other parts of the full LIGHTNING, can be switched off independantly. DP have not made these routines as fast as possible by using integer arithmetic, since this places further restrictions on compatibility.

Both SPEEDSCREEN and LIGHTNING add extra commands to BASIC to enable you to change the speed of SCROLLing and also to attach fonts to different channels (although LIGHTNING does not include a font editor, this is not really a problem, since many exist on the market or in magazines). SPEEDSCREEN also adds commands to alter the CSIZE of characters beyond those given on the basic QL. LIGHTNING does not offer these, but instead it offers you the option of a NUL channel which means that you can select between a printer output channel or nothing without having to revert to putting IF...THEN statements in every line that prints characters to the printer channel. DP has also provided a 'HOTKEY', which when pressed will turn screen output on or off, without stopping the program. However this does not seem to prevent output for very long on some of my programs, so try it and see.

In conclusion, I believe that both packages are excellent, but feel that LIGHTNING has the edge over SPEEDSCREEN for increased speed. The fact that LIGHTNING also speeds up other areas of the QL also adds to this preference. However, if you only have a BASIC 128K QL the SPEEDSCREEN package is perhaps better for you, since the full LIGHTNING package takes up much more memory than the full SPEEDSCREEN version. (c) It is really a trade off between speed and facilities with memory requirements; if you have enough memory, buy LIGHTNING.

(a) COMPARISON OF TEXT OUTPUT

INK	OVER	UNDER	PAPER	CSIZE	BASIC SPEEDSCREEN LIGHTNING		
					(secs)	1.26n	1.11
7	0	0	0	1	102	13	10
0	0	0	7	1	102	19	19
0	0	0	0	1	102	15	19 (*)
0	0	0	7	0	130	42	41
7	0	0	0	0	130	31	29
0	0	0	0	0	130	34	41 (*)

(*) - Please note that this is not really a fair comparison, since LIGHTNING is a little more sensible than SPEEDSCREEN here. How many times do you want to output black text on a black background? (I have included these figures though, just in case there are some programmers out there who use this in their programs).

(b) - If LIGHTNING is switched on while HACK adventure by W. SOKER from the QUANTA library is EXECuted, all of the output will be black on black paper for some reason, thereby making the game unplayable. However, note that this problem also exists with SPEEDSCREEN. If anyone knows why, and possibly a solution to the problem, please write and tell QLTR.

The LIGHTNING manual mentions that to cure 'problem' programs such as this, the user should issue the lngBAD command, which ought to cure the problem, with only a slight loss in speed. However, this command does not cure the problem in HACK - the manual states that DP was also unable to tame LATTICE 'C' in this way. Since HACK was originally written in C, it is possible that any programs developed with LATTICE 'C' will also encounter this problem.

(c) The following are the lengths of the enhancement packages -

LIGHTNING		SPEEDSCREEN (2)	
Fonts	214 bytes	Full Version	11K
Graphics	4258 bytes	Mini Version	5.5K
Maths	6880 bytes	Full Version With Fat Founts	17.5K
Text (1)	21652 bytes		
Mini Text	4564 bytes		

(1) The full text version of LIGHTNING supports fat founts.

(2) Note there are other versions of speedscreen available which have memory requirements between 5.5K and 17.5K

Rich Mellor.